AD-753 416

COMPUTATIONAL ALGORITHMS FOR UNCONSTRAINED OPTIMIZATION

Bruce T. Kujawski

Air Force Flight Dynamic Laboratory
Wright-Patterson Air Force Base, Ohio

October 1972

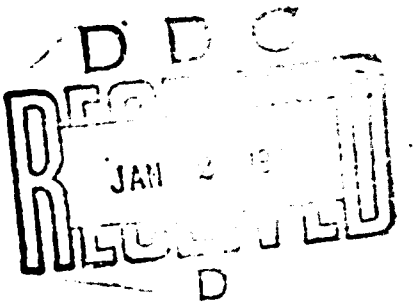
AFFDL-TR-72-77

COMPUTATIONAL ALGORITHMS FOR UNCONSTRAINED OPTIMIZATION

BRUCE T. KUJAWSKI, MAJOR, USAF

TECHNICAL REPORT AFFDL-TR-72-77

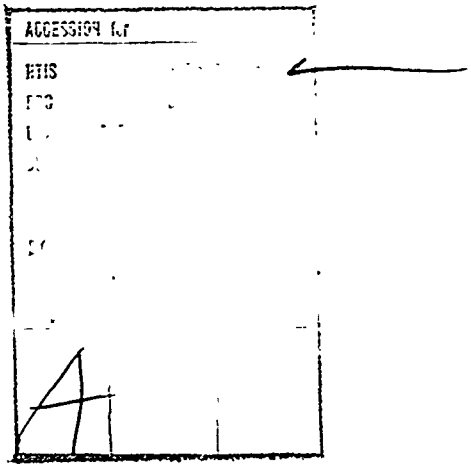
OCTOBER 1972



AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.



Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/7 December 1972 — 300

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Air Force Flight Dynamics Laboratory Wright-Patterson AFB, Ohio 45433	Unclassified
	2b. GROUP

3 REPORT TITLE

COMPUTATIONAL ALGORITHMS FOR UNCONSTRAINED OPTIMIZATION

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5 AUTHOR(S) (First name, middle initial, last name)

Bruce T. Kujawski, Major, USAF

6 REPORT DATE	7a. TOTAL NO OF PAGES	7b. NO. OF REFS
October 1972		15

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO 8219	AFFDL-TR-72-77
c. Task No. 821911	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10 DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY
Air Force Institute of Technology DS/EE/72-1	Air Force Flight Dynamics Laboratory Air Force Systems Command Wright-Patterson AFB, Ohio 45433

13 ABSTRACT

A generalized descent algorithm theory is developed for unconstrained minimization problems. Here a descent algorithm is defined as a computational procedure where at each iteration a descent direction is determined and a single-dimensional search is made for the minimum in the descent direction. The theory is shown to be a generalization of the three most common descent algorithms; gradient, conjugate gradient, and Fletcher-Powell.

Execution of the single-dimensional search can be computationally time consuming. Two additional algorithms are presented which reduce or eliminate single-dimensional search time. The first is a modification of Davidon's Variance Algorithm and requires a minimal single-dimensional search. The second is a direct method for minimizing a special class of quadratic functions.

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Descent Algorithms						
Function Minimizations						
Rank-One						
Conjugate Gradient						
Fletcher-Powell Method						
Davidon Variance						
Algorithms						

*U.S.Government Printing Office: 1972 — 759-488/213

AFFDL-TR-72-77

COMPUTATIONAL ALGORITHMS FOR UNCONSTRAINED OPTIMIZATION

BRUCE T. KUJAWSKI, MAJOR, USAF

FOREWORD

This report was originally prepared as a dissertation in partial fulfillment of the requirement for the degree of Doctor of Philosophy in Aerospace Engineering from the Air Force Institute of Technology, Wright-Patterson AFB, Ohio by Captain Bruce T. Kujawski. Capt. Kujawski's advisor was Lt. Colonel Russell Hannen.

The research was accomplished while the author was assigned to the Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio. The research was conducted under Project 8219, Task 821911, "Flight Control Optimization Techniques".

The author gratefully acknowledges the assistance, suggestions, and stimulations provided by many people in the Laboratory, particularly Mr. Ronald O. Anderson, Capt. Don C. Eckholdt, and Mr. Jerry E. Jenkins.

Special acknowledgement is given to Dr. Donald O. Norris, formerly of AFIT, and presently at the Ohio University, for his patience, suggestions, and encouragement.

This technical report has been reviewed and is approved.

C. B. WESTBROOK
Chief, Control Criteria Branch
Flight Control Division
AF Flight Dynamics Laboratory

ABSTRACT

A generalized descent algorithm theory is developed for unconstrained minimization problems. Here a descent algorithm is defined as a computational procedure where at each iteration a descent direction is determined and a single-dimensional search is made for the minimum in the descent direction. The theory is shown to be a generalization of the three most common descent algorithms; gradient, conjugate gradient, and Fletcher-Powell.

Execution of the single-dimensional search can be computationally time consuming. Two additional algorithms are presented which reduce or eliminate single-dimensional search time. The first is a modification of Davidon's Variance Algorithm and requires a minimal single-dimensional search. The second is a direct method for minimizing a special class of quadratic functions of the form $1/2||x||^2 + 1/2\ k(a - m'x)^2$.

TABLE OF CONTENTS

SYMBOLS

$\frown$	belongs to, is a member of; $x \frown X$: x is a member of the set X.
$\rightarrow$	approaches, converges to
$\downarrow$	monotonically decreasing; $f(x_n) \downarrow L$: $f(x_n)$ monotonically decreases to the value L
$[y,z]$	value of the continuous linear functional y operating on the vector z. In a Hilbert space this is the inner product of the vector y and z. Also, a closed interval on the real line, alternate to parenthesis, and references; context will make usage clear.
$\|\|x\|\|$	The norm of the vector x
$\cup$	The union of sets
$\subseteq$	is a subset of (or equal to)
$'$	matrix or vector transpose except in Section II where f' is the derivative of f.
R^N	Euclidean N-space
D^2	second differential operator
$\{M\}_{ij}$	element of the i^{th} row and j^{th} column of the matrix M

SECTION I

INTRODUCTION

1. BACKGROUND

In virtually all fields of the physical sciences, and particularly in engineering, the digital computer is the principal tool used in the solution of complex problems. The speed and flexibility of the computer has in many cases changed the nature of the problems that can be solved, i.e., the solution must not only meet specific constraints but must also be the best or optimal in some specified sense. There are three interrelated tasks in the formulation of such an optimization problem.

First, the physical system or process must be described mathematically or modeled in terms appropriate for computation. Second, the measure of goodness, generally referred to as the cost function, penalty function or payoff function must be defined to adequately describe how one solution compares to another. Finally, computational methods must be applied to find a solution which satisfies the mathematical model and cost function in such a way so as to extract the best or optimum solution.

Generally such a problem can be cast into a constrained optimization problem such as: Find the solution x which minimizes the cost function $f(x)$ while satisfying specific constraints described by $g(x)=0$. Often the problem can be simplified, conceptually, by adjoining the constraints to the cost function through the use of Lagrange multipliers. Thus the constrained optimization problem is converted

to the following unconstrained problem: Find the solution (x,λ) which extremizes the cost function $F(x,\lambda)=f(x)+\lambda g(x)$. The existence of such Lagrange multipliers is a subject in itself.

Another method of solving the constrained optimization problem is to restrict the problem to a subspace, an approximation to the constraints for example, and considering a related unconstrained problem as an intermediate step in obtaining the solution to the constrained problem. Since the simplified unconstrained problem may have to be solved many times in order to obtain the solution of constrained problem, an efficient method of solving the unconstrained problem is essential. Finally, the solution of the unconstrained problem is often of interest in itself.

The subject of this thesis is the computational methods which may be used to arrive at a minimizing solution to the unconstrained minimization problem. It is tacitly assumed that any constraints are accounted for through the use of Lagrange multipliers or other valid techniques, such as penalty factors.

OUTLINE AND PREVIEW OF SECTIONS

For functions which have a continuous first derivative the most common methods used to minimize the function, i.e., obtain the solution to the unconstrained minimization problem, are the gradient, conjugate gradient, and Fletcher-Powell algorithms. These algorithms are reviewed briefly to illustrate certain common elements. Here it is assumed the function to be minimized is f which is defined for each x in some space X. Further, assume the gradient, g, of f at x also exists: $g(x)=\text{grad } f(x)$. For each algorithm only the initializations required and the recursive equations are given. Convergence criteria or tests

for convergence although important in computational applications are omitted here in order to emphasize those properties which these algorithms have in common.

Gradient Algorithm:

Initially: choose an arbitrary x_o

Iteratively: set $s_n = -g(x_n)$

choose $\alpha=\alpha_n$ to minimize $f(x_n+\alpha s_n)$

set $x_{n+1}=x_n + \alpha_n s_n$.

Conjugate Gradient Algorithm: (Reference 5)

Initially: choose an arbitrary x_o

set $s_o = -g(x_o)$

Iteratively: choose $\alpha = \alpha_n$ to minimize $f(x_n+\alpha s_n)$

set $x_{n+1} = x_n + \alpha_n s_n$

$$\beta_n = \frac{||g(x_{n+1})||^2}{||g(x_n)||^2}$$

$$s_{n+1} = -g(x_{n+1}) + \beta_n s_n.$$

Fletcher-Powell Algorithm: (Reference 4)

Initially: choose an arbitrary x_o

set $H_n = I$

Iteratively: $s_n = -H_n g(x_n)$

choose $\alpha = \alpha_n$ to minimize $f(x_n+\alpha s_n)$

set $x_{n+1} = x_n + \alpha_n s_n$

$$\sigma_n = x_{n+1} - x_n$$

$$y_n=g(x_{n+1}) - g(x_n)$$

$$H_{n+1}=H_n - \frac{H_n y_n y_n' H_n}{y_n' H_n y_n} + \frac{\sigma_n \sigma_n'}{\sigma_n' y_n}$$

where the prime (') denotes transpose.

Each of these algorithms generates a search direction, s_n, for which the function, initially at least, tends to decrease, i.e., for which $g'(x_n)s_n<0$. A single-dimensional search is then conducted to obtain the minimum of f in the direction s_n from the current point x_n. The location of the minimum of the single-dimensional search is chosen as the next iteration point, x_{n+1}. The differences between the algorithms are in the method used to generate the search directions s_n. These algorithms and others which generate a descent direction and incorporate a single-dimensional search will be collectively classed as descent algorithms.

In Section II the proof of a theorem which is a generalization of descent algorithms is presented. Specific applications to the gradient, Fletcher-Powell, and conjugate gradient algorithms are given at the end of the chapter.

Next, consider the problem of minimizing the quadratic function $f(x) = f_o + a'x + 1/2x'Gx$. The gradient of f at x is given by $g(x) = a + Gx$. Let $h = -G^{-1}g(x)$, assuming G^{-1} exists, then

$$\begin{aligned} g(x+h) &= a + G(x+h) \\ &= a + Gx + Gh \\ &= g(x) - g(x) \\ &= 0 \end{aligned}$$

that is, $x^* = x+h$ satisfies $g(x^*) = 0$, the necessary condition for f to have a minimum at x^*. Note that G is the second derivative of f. Clearly for a quadratic function, knowledge of G or the second derivative, or better G^{-1}, greatly simplifies the problem of finding the minimum of the function. Since many functions can be approximated by a

quadratic in some neighborhood of a (local) minimum, information about the second derivative, or its inverse, should enhance the ability to arrive at a solution to the general unconstrained minimization problem.

The Newton-Raphson algorithm is a method of function minimization which utilizes the inverse of the second derivative. In addition to the computational difficulties of obtaining the second derivative, this method requires an initial estimate sufficiently close to the final solution before convergence is guaranteed. Because of these difficulties, several algorithms termed quasi-Newton methods by Powell (Reference 12 have been constructed which iteratively estimate the inverse of the second derivative. The best known of these is the method of Fletcher-Powell. Another more recent method of this type is Davidon's Variance Algorithm, not to be confused with Davidon's Variable Metric Algorithm which was the predecessor to Fletcher-Powell's method. A new derivation of Davidon's Variance Algorithm is presented in Section III. Davidon's algorithm suffers some difficulties and Section III concludes with a modified version of the algorithm which, although somewhat more complex, circumvents one major difficulty.

Whenever a method is available for obtaining the inverse of the second derivative, particularly for a quadratic function, the minimizing solution can be obtained directly. In Section IV a method for the direct solution of a special class of quadratic minimization problems is presented. The procedure is based on the Rank-One method of matrix inversion. The algorithm contains a necessary and sufficient test for the existence of an extremum and a sufficient test that the extremum be a minimum. The special class of problems to which this method applies are generalizations of the following form: $f(x) = 1/2||x||^2 + k(a-m'x)^2$.

SECTION II

DESCENT ALGORITHMS

In this section the proof of a basic theorem on descent algorithms is presented followed by applications to several familiar descent algorithms.

1. NOTATION AND DEFINITIONS

Suppose X denotes a real normed linear space and f a real-valued function defined on X. For an arbitrary point x_0 of X, denote by S the "level set" of f at x_0, i.e., $S = \{x : f(x) \leq f(x_0)\}$. The Frechet derivative of f at x will be denoted $f'(x)$ and if $x^* \in X^*$, the topological dual of X, the value of x^* at x will be written $[x^*, x]$.

Let ϕ denote a bounded map from S to X satisfying:

(i) $[f'(x), \phi(x)] \geq 0$ for all $x \in S$, and

(ii) given an $\varepsilon > 0$ there exists $\delta > 0$ such that

$$[f'(x), \phi(x)] < \delta \text{ implies } ||f'(x)|| < \varepsilon.$$

Observe for later reference that condition (ii) implies that $f'(x) = 0$ whenever $[f'(x), \phi(x)] = 0$ for if there exists an x_1 such that $[f'(x_1), \phi(x_1)] = 0$ but $f'(x_1) \neq 0$ set $\varepsilon = 1/2||f'(x_1)|| > 0$ then $[f'(x_1), \phi(x_1)] = 0 < \delta$ for all δ while $||f'(x_1)|| = 2\varepsilon > \varepsilon$ contrary to (ii). Condition (ii) also implies $[f'(x), \phi(x)]$ is bounded away from zero whenever $f'(x)$ is bounded away from zero in the following sense. If for $\{x_{n_k}\} \subset S$, $||f'(x_{n_k})|| > \varepsilon$ for some $\varepsilon > 0$; then there exists a subsequence of $\{x_{n_k}\}$ and $\delta > 0$ such that $[f'(x_{n_k}), \phi(x_{n_k})] \geq \delta$.

In the theorem which follows, $-\phi(x)$ serves to define the descent direction.

2. BASIC THEOREM ON GENERALIZED DESCENT ALGORITHMS

Many of the ideas in the following theorem, were stimulated by two papers by A. A. Goldstein (References 6, 7). In particular the definition of ϕ given above and the form of the conclusions of the theorem are identical to those of Goldstein. The hypotheses of the theorem are changed to specialize to the case of a single-dimensional search at each iteration. Thus the proof of part (a) of the theorem is changed. The proofs of parts (b) and (c) follow Goldstein. The following additional remark on the hypotheses is in order.

In the current setting where X is a normed linear space, the assumption that f' is uniformly continuous on S may be replaced with the equivalent conditions that f is uniformly differentiable and that f' is bounded on S (Reference 14, p. 45).

Theorem I

Assume S is bounded in X, f is bounded below on S and the Frechet-derivative f' of f exists, is uniformly continuous on S and bounded on S.

Set $x_{n+1} = x_n$ when $[f'(x_n), \phi(x_n)] = 0$, otherwise choose $\rho=\rho_n$ to minimize $\{f(x_n - \rho\phi(x_n)):\rho\geq 0\}$ and set $x_{n+1} = x_n - \rho_n\phi(x_n)$. Then

(a) $f(x_n) \to L$, $f'(x_n)\to 0$,

(b) if $\{x_n\}$ has cluster points, every cluster point z satisfies $f(z)=L$, $f'(z)=0$.

(c) If f' has finitely many zeros on S, S is compact, and $||x_{n+1} - x_n||\to 0$, then the sequence $\{x_n\}$ converges.

Proof

(a) For $x\in S$ and $f'(x) \neq 0$, $[f'(x), \phi(x)]>0$. In general, since f is differentiable,

$$[f'(x), h] = \lim_{t\to 0} \frac{1}{t}(f(x+th) - f(x))$$

$$= \lim_{t\to 0^-} \frac{1}{t}(f(x+th) - f(x))$$

$$= \lim_{\rho\to 0^+} \frac{1}{-\rho}(f(x-\rho h) - f(x)),$$

so in particular,

$$0 < [f'(x), \phi(x)] = \lim_{\rho\to 0^+} \frac{1}{-\rho}(f(x-\rho\phi(x)) - f(x)).$$

By the uniform differentiability of f there exists a $\rho_o>0$, independent of x, such that

$$0<\frac{1}{-\rho_o}(f(x-\rho_o\phi(x)) - f(x))$$

so,

$$0>f(x-\rho_o\phi(x)) - f(x)$$

or

$$f(x-\rho_o\phi(x))<f(x).$$

In particular for $x = x_n \in S$, $f(x_n - \rho_o\phi(x_n))<f(x_n)\leq f(x_o)$.

Also for $f'(x) \neq 0$, $[f'(x), \phi(x)] \neq 0$, hence $||\phi(x)||>0$. Since S is assumed bounded, there exists a $\rho_a>0$ such that for all $\rho>\rho_a$, $x - \rho\phi(x) \notin S$. For on the contrary assumption, for every N, no matter how large, there exists a $\rho_N>N$ such that $y = x - \rho_N\phi(x)\in S$. Then $||y-x|| = \rho_N||\phi(x)||>N||\phi(x)||$ is unbounded which contradicts the boundedness of S.

Now for $x_n \in S$ such that $f'(x_n) \neq 0$, $F(\rho) = f(x_n - \rho\phi(x_n)) - f(x_n)$ has a minimum at $\rho_n\in[0, \rho_a]$ since F is continuous and $[0, \rho_a]$ is compact. Furthermore, $\rho_n \neq 0$ since $F(\rho_o) = f(x_n - \rho_o\phi(x_n)) - f(x_n)<F(0)$. Thus the sequences $\{\rho_n\}$ and $\{x_n\}$ are well defined and $f(x_{n+1}) = f(x_n - \rho_n\phi(x_n))<f(x_n)$ which implies the sequence $\{x_n\}\subseteq S$, and the sequence

$\{f(x_n)\}$ is strictly decreasing. Since f is assumed bounded below $f(x_n) \downarrow L$.

To show $f'(x_n) \to 0$, we suppose the contrary, then there exists a subsequence $\{x_{n_k}\} \subseteq \{x_n\}$ such that $f'(x_{n_k})$ is bounded away from zero, which implies that a subsequence of $[f'(x_{n_k}), \phi(x_{n_k})]$ is bounded away from zero. Without loss of generality denote this subsequence by $\{x_{n_k}\}$. Then there exists an $\epsilon_o > 0$ such that $\epsilon_o < [f'(x_{n_k}), \phi(x_{n_k})] = \lim_{\rho \to 0^+} \frac{1}{-\rho} (f(x_{n_k} - \rho\phi(x_{n_k})) - f(x_{n_k}))$. Since f is uniformly differentiable, there exists a $\rho_b > 0$, independent of x (i.e., x_{n_k}), such that

$$\frac{\epsilon_o}{2} < \frac{1}{-\rho_b} (f(x_{n_k} - \rho_b\phi(x_{n_k})) - f(x_{n_k})),$$

$$\frac{-\rho_b\epsilon_o}{2} > f(x_{n_k} - \rho_b\phi(x_{n_k})) - f(x_{n_k}),$$

or $$f(x_{n_k} - \rho_b\phi(x_{n_k})) < f(x_{n_k}) - \frac{\rho_b\epsilon_o}{2} .$$

Then $f(x_{n_k+1}) \leq f(x_{n_k} - \rho_b\phi(x_{n_k})) < f(x_{n_k}) - \rho_b\frac{\epsilon_o}{2}$.

Since $f(x_n) \downarrow L$,

$$f(x_{n_k}) < f(x_{n_k-1}) < \cdots < f(x_{n_{k-1}+1}) < f(x_{n_{k-1}}) - \frac{\rho_b\epsilon_o}{2}$$

that is,

$$\begin{aligned} f(x_{n_k}) &< f(x_{n_{k-1}}) - \frac{\rho_b\epsilon_o}{2} , \\ &< f(x_{n_{k-2}}) - 2\frac{\rho_b\epsilon_o}{2} , \\ &< f(x_{n_{k-3}}) - 3\frac{\rho_b\epsilon_o}{2} , \\ &\cdots \\ &< f(x_{n_o}) - k\frac{\rho_b\epsilon_o}{2} . \end{aligned}$$

This contradicts the assumption that f is bounded below, hence $f'(x_n) \to 0$.

(b) If z is a cluster point of $\{x_n\}$ there exists a subsequence $\{x_{n_k}\} \subseteq \{x_n\}$ such that $x_{n_k} \to z$ (in norm). Since f and f' are continuous, $f(x_n) \to L$ and $f'(x_n) \to 0$, it follows that $f(z) = L$ and $f'(z) = 0$.

(c) Since $f'(z) = 0$ for every cluster point z of $\{x_n\}$, the number of roots of f' on S is equal to or greater than the number of cluster points of $\{x_n\}$. If f' has a unique root z on S then $\{x_n\}$ converges to it; for otherwise, since S is assumed compact, there exists at least one cluster point z_1 of $\{x_n\}$ in S where $f'(z_1) = 0$ (by (b)). If $z_1 \neq z$, the root of f' is not unique. If the number of roots of f' on S is finite then the number of cluster points of $\{x_n\}$ is finite also.

Let z_i, $i = 1,2,\ldots,k$ be the cluster points of $\{x_n\}$, let $\varepsilon = \min\{||z_i - z_j||: i \neq j, i,j = 1,2,\ldots k\}$, let $S(z_i,\varepsilon/3)$ denote the open sphere of radius $\varepsilon/3$ centered at z_i. Since S is assumed compact the set $\{x_n\} - \bigcup_{i=1}^{k} S(z_i,\varepsilon/3)$ contains a finite number of points, say m.

Since $||x_{n+1} - x_n|| \to 0$, by assumption, there exists an N such that $||x_{p+1} - x_p|| < \varepsilon/3m$ for all $p>N$.

Now, since the z_i, $i=1,\ldots k$ are cluster points of $\{x_n\}$, there are members of $\{x_n\}$ in each $S(z_i,\varepsilon/3)$ for which $n>N$. Therefore fix $n>N$ such that $x_n \in S(z_i, \varepsilon/3)$, for some fixed i, and $x_{n+1} \notin S(z_i,\varepsilon/3)$. Let x_q be the next member of the sequence $\{x_n\}$ (i.e., $q>n$) such that $x_q \in S(z_j, \varepsilon/3)$, for some $j \neq i$ and $x_{q-1} \notin S(z_j,\varepsilon/3)$.

Since $x_n \in S(z_i,\varepsilon/3)$ and $x_q \in S(z_j, \varepsilon,3)$, $i \neq j$, $||x_n - x_q|| > \varepsilon/3$. On the other hand, since n, $q>N$, we have

$$||x_q - x_n|| \leq ||x_q - x_{q-1}|| + ||x_{q-1} - x_{q-2}|| + \cdots + ||x_{n+2} - x_{n+1}|| + ||x_{n+1} - x_n|| = \sum_{p=n}^{q-1} ||x_{p+1} - x_p|| < \sum_{p=n}^{q-1} \frac{\varepsilon}{3m} = (q-n)\frac{\varepsilon}{3m}.$$

Then $\frac{\epsilon}{3} < \|x_q - x_n\| < (q-n)\frac{\epsilon}{3m}$ or $m<q-n$.

Now the set of points $x_{n+1}, x_{n+2}, \cdots, x_{q-1}$ belong to the set $\{x_n\} - \bigcup_{i=1}^{k} S(z_i, \epsilon/3)$. But there are $(q-1) - (n+i) + i = q-n-1$ points in the set $\{x_{n+1}, \cdots, x_{q-1}\}$, and by supposition m points in the set $\{x_n\} - \bigcup_{i=1}^{k} S(z_i, \epsilon/3)$. Thus $m \geq q-n-1$. On the other hand we have just shown $q-n>m$, thus $m \geq q-n-1 > m-1$, that is to say $q-n-1 = m$. Heuristically, all the points of the set $\{x_n\} - \bigcup_{i=1}^{k} S(z_i, \epsilon/3)$ have been accounted for, or "used up". But by the same argument, there exists an $n' \geq q$ and a $q' > n'$ such that $x_{n'} \in S(z_j, \epsilon/3)$ and $x_{n'+1} \notin S(z_j, \epsilon/3)$, for the j defined by q above (i.e., for which $x_q \in S(z_j, \epsilon/3)$), and $x_{\zeta'} \in S(z_\ell, \epsilon/3)$, $\ell \neq j$, and $x_{q'-1} \notin S(z_\ell, \epsilon/3)$. Then there are again $q'-n'-1 = m$ points in the set $\{x_{n'+1}, x_{n'+2}, \cdots, x_{q'-1}\}$ which also belong to the set $\{x_n\} - \bigcup_{i=1}^{k} S(z_i, \epsilon/3)$. There now are 2m points in this set which by supposition contained only m. This contradiction persists unless we suppose $\{x_n\}$ has a unique cluster point.

Comments on the Basic Theorem

The first conclusion of the theorem, $f(x_n) \to L$, can be obtained with the weaker assumption that f is differentiable and without the assumption that ϕ is bounded, in which case both ρ_o and ρ_a are dependent on x (or x_n). However, both uniform differentiability of f, which follows from the uniform continuity of f', and boundedness of ϕ were used to assure the existence of a ρ_b which uniformly bounds the differential away from zero in the proof leading to $f'(x_n) \to 0$.

If X is finite dimensional then S being closed and bounded is necessarily compact and $\{x_n\} \subseteq S$ has cluster points. Hence conclusion (b) of the theorem applies.

3. APPLICATION OF THE BASIC THEOREM TO COMMON DESCENT ALGORITHMS

In the applications which follow, the boundedness of ϕ will be derived from the boundness of the Frechet derivative f' of f on S. In particular, assume now that X is a Hilbert space so that f'(x) can be represented by its gradient $\nabla f(x)$ in X. Then also since in a Hilbert space every bounded set is weakly compact, the boundedness of f' on S follows from its continuity on S (Reference 3, p. 19).

Corollary I

Let Q be a positive definite continuous linear operator on X, let $\phi(x) = Q\nabla f(x)$, then Theorem I applies.

Proof:

Since f has the required properties, all that remains is to show ϕ is bounded and satisfies conditions (i) and (ii).
By assumption f' is uniformly continuous and bounded on S. This together with Q being a continuous linear operator on X implies $\phi = Q\nabla f$ is bounded on S.

Q positive definite implies there exists a $m>0$ such that $m||z||^2 \leq [z, Qz]$ for all z in X,
hence

$[f'(x), \phi(x)] = [f'(x), Qf'(x)] \geq m||f'(x)||^2 \geq 0$ and condition (i) is satisfied.

To show condition (ii) assume an $\varepsilon>0$ and choose $\delta=m\varepsilon^2$. Then $[f'(x), \phi(x)] < \delta=m\varepsilon^2$ implies $m||\nabla f(x)||^2 \leq [\nabla f(x), Q\nabla f(x)] < \delta=m\varepsilon^2$ or $||\nabla f(x)|| < \varepsilon$.

In order to demonstrate that the theory developed thus far is a generalization of descent algorithms, the Theorem and Corollary will be applied to two common algorithms; the gradient and the Fletcher-Powell methods.

Application to Gradient Algorithms

Let Q in Corollary I be the identity operator, i.e., $\phi(x) = \nabla f(x)$, this is the usual gradient algorithm.

The operator Q need not remain fixed so long as it is uniformly positive definite on S. That is to say there exists a constant $m>0$, independent of x, such that $[f'(x), \phi(x)] = [f'(x), Q(x)\nabla f(x)] \geq m||\nabla f(x)||^2$ for all x in S.

Application to Fletcher-Powell

Let X be finite dimensional, let $Q = Q(k) = H_k$ as defined by R. Fletcher and M.J.D. Powell in Reference 4. Using the algorithm of Fletcher-Powell it is easy to prove H_k remains positive definite at each iteration unless the algorithm terminates in a finite number of steps. Then $\phi(x_n) = H_n\nabla f(x_n)$ satisfies condition (i): $[f'(x_n), \phi(x_n)] \geq 0$. Now if the H_k are uniformly positive definite, or if there exist constants $m>0$ and $p>0$, independent of k, such that $[x, H_k x] \geq m||x||^p$, then condition (ii) is satisfied. For given an $\varepsilon>0$, choose $\delta=m\varepsilon^p$, then $m||f'(x)||^p \leq [f'(x), H^k f'(x)] < \delta=m\varepsilon^p$ implies $||f'(x)||<\varepsilon$. Then the theorem applies to the Fletcher-Powell algorithm. Furthermore, since S is closed and bounded, it is compact, and the sequence $\{x_k\}$ necessarily has cluster points so (b) of the theorem applies without further assumption.

In the above application to Fletcher-Powell's method and in the following corollary, the function ϕ is not uniquely defined as a

function from S into X. Rather, ϕ is defined only on the sequence $\{x_n\}$ and its value at an arbitrary x in S depends upon which sequence x belongs to. However, the properties of ϕ used in the proof of Theorem I did not depend on ϕ being defined anywhere except at points of the sequence $\{x_n\}$. That is to say the conditions (i) and (ii) need hold only at the points of the sequence $\{x_n\}$. Within this context the search direction will continue to be denoted by $-\phi(x_n)$.

Corollary II

Let $\phi(x_o) = \nabla f(x_o)$ and

$$\phi(x_n) = \nabla f(x_n) + K_n \phi(x_{n-1}) \text{ for } n = 1,2...$$

where $|K_n| \leq r < 1$, then Theorem I applies.

Proof:

Clearly conditions (i) and (ii) are satisfied at x_o for $[f'(x_o), \phi(x_o)] = ||\nabla f(x_o)||^2 \geq 0$, and given $\varepsilon > 0$ choose $\delta = \varepsilon^2$.

Claim $[f'(x_{n+1}), \phi(x_n)] = 0$ for $n = 0,1,....$ Since ρ_n is chosen to minimize $\{F(\rho) = f(x_n - \rho\phi(x_n)) - f(x_n): \rho > 0\}$, $F'(\rho_n) = 0$. But $F'(\rho) = -[f'(x_n - \rho\phi(x_n)), \phi(x_n)]$ and at $\rho = \rho_n$, $x_n - \rho_n\phi(x_n) = x_{n+1}$. Hence $F'(\rho_n) = -[f'(x_{n+1}), \phi(x_n)] = 0$.

Then for $n = 1,2,...$

$$[f'(x_n), \phi(x_n)] = [f'(x_n), \nabla f(x_n)] + K_n[f'(x_n), \phi(x_{n-1})]$$
$$= [\nabla f(x_n), \nabla f(x_n)]$$
$$= ||\nabla f(x_n)||^2 > 0.$$

Hence again, given $\varepsilon > 0$ choose $\delta = \varepsilon^2$ then $[f'(x_n), \phi(x_n)] < \delta$ implies $||\nabla f(x_n)|| = ||f'(x_n)|| < \varepsilon$. Thus conditions (i) and (ii) are satisfied.

To show boundedness of the $\phi(x_n)$ consider

$$||\phi(x_n)||^2 = [\phi(x_n), \phi(x_n)] = [\nabla f(x_n), \phi(x_n)] + K_n[\phi(x_{n-1}), \phi(x_n)]$$
$$= ||\nabla f(x_n)||^2 + K_n[\phi(x_{n-1}), \phi(x_n)], \text{ and}$$

$$[\phi(x_n),\ \phi(x_{n-1})] = [\nabla f(x_n),\ \phi(x_{n-1})] + K_n[\phi(x_{n-1}),\ \phi(x_{n-1})]$$
$$= K_n\ ||\phi(x_{n-1})||^2.$$

Thus, $||\phi(x_n)||^2 = ||\nabla f(x_n)||^2 + K_n^2||\phi(x_{n-1})||^2$.

Applying this relation recursively yields

$$||\phi(x_n)||^2 = ||\nabla f(x_n)||^2 + K_n^2\{||\nabla f(x_{n-1})||^2 + K_{n-1}^2||\phi(x_{n-2})||^2\}$$
$$= ||\nabla f(x_n)||^2 + K_n^2||\nabla f(x_{n-1})||^2 + K_n^2K_{n-1}^2||\phi(x_{n-2})||^2$$
$$\cdots$$
$$||\phi(x_n)||^2 = ||\nabla f(x_n)||^2 + K_n^2||^2f(x_{n-1})||^2 + K_n^2K_{n-1}^2||\nabla f(x_{n-2})||^2$$
$$+ \cdots + K_n^2K_{n-1}^2 \cdots K_2^2||\nabla f(x_1)||^2$$
$$+ K_n^2K_{n-1}^2 \cdots K_2^2K_1^2||\nabla f(x_o)||^2.$$

Since $f'(x)$ is bounded for $x \in S$ let $M = \sup_{x \in S} ||\nabla f(x)||$

then

$$||\phi(x_n)||^2 \leq \{1 + K_n^2 + K_n^2K_{n-1}^2 + \cdots + K_n^2K_{n-1}^2 \cdots K_2^2K_1^2\}M^2.$$

Also $K_n^2 \leq r < 1$, so the series

$$\{1 + K_n^2 + K_n^2K_{n-1}^2 + \cdots + K_n^2K_{n-1}^2 \cdots K_2^2K_1^2\}$$

converges. Hence $||\phi(x_n)||^2$ is bounded and therefore $\phi(x_n)$ is bounded.

For a trivial application of Corollary II, K_n may be set to zero for all n, this then generates the usual gradient algorithm. The following application is of much more interest.

Application to Conjugate Gradient

Let $K_n = \dfrac{[\nabla f(x_n), \nabla f(x_n)]}{[\nabla f(x_{n-1}), \nabla f(x_{n-1})]}$ in Corollary II.

This is the β_{n-1} of Fletcher-Reeves (Reference 5) for $X = R^n$, and Lasdon, Mitter and Warren (Reference 10), for X a function space.

The condition $|K_n| \leq r < 1$ may appear overly restrictive particularly in the light of the paper by Lasdon, Mitter and Warren which also is set in a Hilbert space H. Unfortunately, the proof of Lasdon, Mitter and Warren contains one minor but signficant discrepancy. The proof of their theorem is reproduced and corrected, to illustrate the similarities of the constraints which must be imposed on K_n or β_{n-1}.

Theorem 3 of Lasdon, Mitter and Warren

If: 1. J(u) is bounded below,

2. J(u) and g(u) = grad J(u) are continuous,

3. $D^2J(u,h,h)$ exists and $|D^2J(u,h,h)| \leq m||h||^2$ for $m > 0$ and all u,h in H,

4. $\{u_k\}$ has a cluster point u*,

then the sequence $\{u_k\}$ formed with arbitrary u_o by applying the conjugate gradient algorithm to J(u) has the following properties:

1. $\lim_{k\to\infty} J(u_k) = J(u^*)$,

2. $\lim_{k\to\infty} g(u_k) = g(u^*) = 0$.

Before presenting the proof the following three remarks are pertinent:

1. Here $D^2J(u,h,k)$ is the second differential of J at u.

2. The form of the algorithm is exactly as given in Section I.2 except here the independent variable is u instead of x.

3. From the above proof of Corollary II it follows that $[g_k,s_k] = -||g_k||^2$ and $[s_k,g_{k+1}] = 0$ for all k.

Corrected Proof of Theorem 3 of Lasdon, Mitter and Warren

From Taylor's Theorem,

$$J(u_k + \alpha s_k) = J(u_k) + \alpha[g_k,s_k] + 1/2\alpha^2 D^2J(\xi_k(\alpha),s_k,s_k)$$

where $\xi(\alpha)$ belongs to the line segment joining u_k and $u_k+\alpha s_k$. Then, using $[s_k,g_k] = -||g_k||^2$ and assumption 3,

$$J(u_k+\alpha s_k) \leq J(u_k) - \alpha||g_k||^2 + 1/2\alpha^2 m||s_k||^2.$$

Since $\alpha=\alpha_k$ minimizes $J(u_k + \alpha s_k)$,

$$J(u_k + \alpha_k s_k) = J(u_{k+1}) \leq J(u_k + \frac{1}{m}s_k)$$

$$\leq J(u_k) - \frac{1}{m}||g_k||^2 + \frac{1}{2m}||s_k||^2.$$

At this point Lasdon, Mitter and Warren assume $||s_k|| = ||g_k||$ in their proof of this theorem. This is clearly inconsistent with $[s_k,g_k] = -||g_k||^2$ for by the Cauchy-Schwarz inequality $|[s_k,g_k]| \leq ||s_k||\ ||g_k||$ where equality holds if and only if s_k is a multiple of g_k. Thus $||g_k||^2 = |[s_k,g_k]| \leq ||s_k||\ ||g_k||$ or $||g_k|| \leq ||s_k||$. Specifically, equality holds only when k=0 (i.e., $s_o = -g_o$) or $g_k = 0$ in which case the solution has been achieved.

Proceeding more carefully, observe that $s_k = -g_k + \beta_{k-1}s_{k-1}$,

$$||s_k||^2 = [s_k,s_k] = -[s_k,g_k] + \beta_{k-1}[s_k,s_{k-1}]$$

$$= ||g_k||^2 + \beta_{k-1}[s_k,s_{k-1}]$$

and $[s_{k-1},s_k] = -[s_{k-1},g_k] + \beta_{k-1}[s_{k-1},s_{k-1}]$

$$= \beta_{k-1}[s_{k-1},s_{k-1}],$$

or $||s_k||^2 = ||g_k||^2 + \beta_{k-1}^2\ ||s_{k-1}||^2$.

Then

$$J(u_{k+1}) \leq J(u_k) - \frac{1}{m}||g_k||^2 + \frac{1}{2m}(||g_k||^2 + \beta_{k-1}^2\ ||s_{k-1}||^2)$$

$$= J(u_k) - \frac{1}{2m}||g_k||^2 + \frac{1}{2m}\beta_{k-1}^2\ ||s_{k-1}||^2,$$

Likewise,

$$J(u_k) \leq J(u_{k-1}) - \frac{1}{2m}||g_{k-1}||^2 + \frac{1}{2m}\beta_{k-1}^2\ \ ||s_{k-2}||^2,$$

hence

$$J(u_{k+1}) \leq J(u_{k-1}) - \frac{1}{2m}||g_{k-1}||^2 + \frac{1}{2m}\beta_{k-2}^2||s_{k-2}||^2$$

$$- \frac{1}{2m}||g_k||^2 + \frac{1}{2m}\beta_{k-1}^2||s_{k-1}||^2,$$

or in general,

$$J(u_{k+1}) \leq J(u_j) - \sum_{i=j}^{k} \frac{1}{2m}||g_i||^2 + \sum_{i=j-1}^{k-1} \frac{1}{2m}\beta_i^2||s_i||^2,$$

or

$$J(u_k) \leq J(u_o) + \sum_{i=0}^{k-1} \frac{\beta_i^2}{2m}||s_i||^2 - \sum_{i=0}^{k} \frac{1}{2m}||g_i||^2$$

$$= J(u_o) - \sum_{i=1}^{k} z_i - \frac{1}{2m}||g_o||^2,$$

where $z_i = \frac{1}{2m}||g_i||^2 - \frac{\beta_{i-1}^2}{2m}||s_{i-1}||^2$.

Since J(u) is bounded below $\lim_{k\to\infty} \sum_{i=1}^{k} z_i$ exists and is finite, that is to say the series $\sum z_i$ converges.

Since $\sum z_i = \sum\left[\frac{1}{2m}||g_i||^2 - \frac{\beta_{i-1}^2}{2m}||s_{i-1}||^2\right]$ it follows that (Reference 13) if the series $\sum z_i$ is absolutely convergent, then each series $\sum\frac{1}{2m}||g_i||^2$ and $\sum\frac{\beta_{i-1}^2}{2m}||s_{i-1}||^2$ is convergent, which implies $||g_i|| \to 0$ and $\beta_{i-1}^2||s_{i-1}||^2 \to 0$. It also follows that $||s_i||^2 = ||g_i||^2 + \beta_{i-1}^2||s_{i-1}||^2 \to 0$ from which it follows that $||s_i||$ is bounded. On the other hand if $\sum z_i$ is conditionally convergent (convergent but not absolutely convergent) then each series $\sum\frac{1}{2m}||g_i||^2$ and $\sum\frac{\beta_{i-1}^2}{2m}||s_{i-1}||^2$ is divergent. Of course it still may happen that $||g_i|| \to 0$ and $\beta_{i-1}^2||s_{i-1}||^2 \to 0$ (and hence s_i is bounded).

One way to assure $\sum z_i$ is absolutely convergent is to assure $z_i \geq 0$, that is $2m\, z_i = ||g_i||^2 - \beta_{i-1}^2||s_{i-1}||^2 \geq 0$. But $||s_i||^2 = ||g_i||^2 + \beta_{i-1}^2||s_{i-1}||^2$, hence $2m\, z_i = 2||g_i||^2 - ||s_i||^2 \geq 0$

requires $||s_i||^2 \leq 2||g_i||^2$ (recall $||g_i||^2 \leq ||s_i||^2$). Now

$$||s_o||^2 = ||g_o||^2,$$

$$||s_1||^2 = ||g_1||^2 + \beta_o^2||s_o||^2$$

$$= ||g_1||^2 \left(1 + \beta_o \frac{||s_o||^2}{||g_o||^2}\right)$$

$$= ||g_1||^2 (1 + \beta_o)$$

since $\beta_o = ||g_1||^2/||g_o||^2$.

Similarly,

$$||s_2||^2 = ||g_2||^2 \left(1 + \beta_1 \frac{||s_1||^2}{||g_1||^2}\right)$$

$$= ||g_2||^2 (1 + \beta_1(1 + \beta_o))$$

$$= ||g_2||^2 (1 + \beta_1 + \beta_1\beta_o)$$

or in general

$$||s_k||^2 = ||g_k||^2 (1 + \beta_k + \beta_k\beta_{k-1} + \cdots + \beta_k\beta_{k-1} \cdots \beta_o).$$

Now if $\beta_i \leq r < 1/2$ for all i,

$$1 + \beta_k + \beta_k\beta_{k-1} + \cdots + \beta_k\beta_{k-1} \cdots \beta_o \leq \frac{1}{1-r} < \frac{1}{1-1/2} = 2$$

Then $||s_k||^2 < 2||g_k||^2$, and $z_k > 0$, and absolute convergence of $\sum z_k$ follows from its convergence. The condition $\beta_i \leq r < 1/2$ is stronger than the condition $|K_n| \leq r < 1$ assumed in Corollary II.

Application to Conjugate Gradient in R^N

The classic proof of the conjugate gradient algorithm for function minimization in finite dimensional spaces (R^N) is based on the Gram-Schmidt Orthogonalization procedure. Originally the method was developed for the solution of systems of linear equations. The extension to the problem of minimizing a quadratic function on R^N is well-known. In theory, the conjugate gradient method finds the minimum of a quadratic function in at most N steps. However in practice, be-

cause of round-off error, N+1 steps are used to obtain the "exact" solution.

Application of the conjugate gradient algorithm to the problem of minimizing an arbitrary function in R^N usually follows the same procedure, viz., conjugate directions are used for N+1 steps and if the minimum has not been obtained, the algorithm is re-started with the last best estimate (x_{N+1}). In terms of Corollary II, this means $K_{N+1} = K_{2N+2} = \cdots = 0$. This condition may be used to impose a somewhat weaker but less instructive condition on the K_n's in Corollary II, specifically

$$(1 + K_N^2 + K_N^2 K_{N-1}^2 + \cdots + K_N^2 K_{N-1}^2 \cdots K_1^2) \leq A$$

$$(1 + K_{2N+1}^2 + K_{2N+1}^2 K_{2N}^2 + \cdots + K_{2N+1}^2 K_{2N}^2 \cdots K_{N+2}^2) \leq A \text{ etc. for}$$

each sub-cycle. Then $||\phi(x_k)||^2 \leq A M^2$ for all k where $||\nabla f(x_1)||^2 \leq M^2$.

As a final remark it should be noted that in Corollary II the condition $|K_n| \leq r<1$ need not be satisfied for all n but only for all n beyond some point. In the conjugate gradient algorithm $K_n = \dfrac{||\nabla f(x_n)||^2}{||\nabla f(x_{n-1})||^2} \leq r < 1$ implies the gradient not only converges to zero but $||\nabla f(x_n)||$ forms (eventually) a strictly monotone decreasing sequence.

SECTION III

A RANK-ONE METHOD OF FUNCTION MINIMIZATION

A particular property possessed by both the method of conjugate gradients and the method of Fletcher-Powell is that either method obtains the minimum of a positive definite quadratic form; viz., minimizes $f(x) = f_o + a'x + 1/2\ x'Gx$, $x \in R^n$, in a finite number of steps excepting round-off errors. This is accomplished in the conjugate gradient algorithm by means of the Gram-Schmidt orthogonalization process (Reference 1). In the method of Fletcher-Powell, this is accomplished by generating the inverse of the matrix G, specifically, $H_n = G^{-1}$. In this section another method for generating G^{-1} is presented which does not require a single-dimensional search for a minimum as do the methods of Fletcher-Powell and conjugate gradients.

Although this author arrived at the method independently, the algorithm is essentially the same as Davidon's variance algorithm (Reference 3). Although Davidon's proofs are valid, they provide little insight for the user on how the algorithm is structured or why the method works.

The derivation presented here clarifies the structure of the algorithm by emphasizing how the structure leads to the desired properties at each iteration. Finally, a minor change is incorporated into Davidon's algorithm which circumvents one difficulty which may be encountered when the algorithm is applied to a computational problem.

1. BASIC RANK-ONE METHOD

In Reference 15, Herbert S. Wilf presents a method for matrix inversion based on the equation $(A + uv')^{-1} = A^{-1} - \dfrac{A^{-1}uv'A^{-1}}{1 + v'A^{-1}u}$

where the prime (') denotes matrix (vector) transpose and it is assumed A^{-1} exists. This equation is easily verified by computing

$$(A + uv')\left(A^{-1} - \frac{A^{-1}uv'A^{-1}}{1 + v'A^{-1}u}\right) = I.$$

Note that $A + uv'$ differs from A by a matrix, uv', of rank one. Therefore, the above inversion technique and the algorithm developed below might well be called Rank-One Methods.

Following Wilf, suppose C is written in the form $C = C_o + \sum_{i=1}^{N} u_i v_i'$ and let C_n be the partial sums $C_n = C_o + \sum_{i=1}^{n} u_i v_i'$ and D_n denote the partial sum inverses

$$D_n = C_n^{-1} = (C_o + \sum_{i=1}^{n} u_i v_i')^{-1}.$$

Then $C_{n+1} = C_n + u_{n+1}v_{n+1}'$

and $D_{n+1} = (C_o + \sum_{i=1}^{n+1} u_i v_i')^{-1}$

$$= (C_n + u_{n+1}v_{n+1}')^{-1}$$

$$= (D_n^{-1} + u_{n+1}v_{n+1}')^{-1}$$

$$= D_n - \frac{D_n u_{n+1} v_{n+1}' D_n}{1 + v_{n+1}' D_n u_{n+1}}.$$

It follows that $D_N = C_N^{-1} = (C_o + \sum_{i=1}^{N} u_i v_i)^{-1} = G^{-1}$ providing all computations may be carried out. Obviously the procedure fails if the denominator term $1 + v_{n+1}' D_n u_{n+1}$ is zero. The implication of such a condition can be deduced from the results of the following Lemma.

Lemma I

Assume A is nonsingular, then the matrix $B = A + uv'$ is singular if and only if $1 + v'A^{-1}u = 0$.

Proof

Necessity: Assume B is singular, then there exists an $x \neq 0$ such that $Bx = Ax + uv'x = 0$. Then $A^{-1}Bx = x + A^{-1}uv'x = 0$ and

$v'A^{-1}Bx = v'x\,(1 + v'A^{-1}u) = 0$. Now $v'x \neq 0$, for then $Bx = Ax = 0$ but A is nonsingular. Therefore, $(1 + v'Au) = 0$.

Sufficiency: Assume $1 + v'Au = 0$ and set $x = A^{-1}u \neq 0$ for otherwise $u = 0$ and $v'A^{-1}u = 0$ which contradicts $1 + v'A^{-1}u = 0$. Then

$$Bx = Ax + uv'x$$
$$= u + uv'A^{-1}u$$
$$= (1 + v'A^{-1}u)\,u$$
$$= 0.$$

2. STRUCTURE OF THE RANK-ONE MINIMIZATION ALGORITHM

An algorithm will be constructed for minimizing the quadratic form $f(x) = f_o + a'x + 1/2\,x'Gx$, $x \in R^N$ and G a positive definite symmetric matrix. The method is based on the above technique for generating G^{-1}. Let $D_o = C_o^{-1}$, where C_o is an arbitrary positive definite symmetric matrix such as the identity I. Pick an arbitrary x_o and let $g_o = a + Gx_o = \text{grad } f(x_o)$. Assume $g_o \neq 0$ for otherwise x_o is the required solution. Set $\sigma_o = -D_o g_o$, $x_1 = x_o + \sigma_o$, $g_1 = \text{grad } f(x_1) = a + Gx_1$ and $y_o = g_1 - g_o$. Then $g_1 = a + G(x_o + \sigma_o) = g_o + G\sigma_o$ and $y_o = G\sigma_o$. An improved estimate C_1 to G is sought such that $C_1\sigma_o = G\sigma_o = y_o$, where C_1 has the form $C_1 = C_o + u_1v_1'$. Since G is symmetric, being the Hessian of f, the added constraint $u_1 = v_1$ is imposed to assure symmetry of C_1. Then the condition to be satisfied is $C_1\sigma_o = C_o\sigma_o + u_1u_1'\sigma_o = y_o$, or $u_1 = \dfrac{y_o - C_o\sigma_o}{u_1'\sigma_o}$. But $-C_o\sigma_o = -C_o(-D_og_o) = g_o$, so $u_1 = \dfrac{g_1}{u_1'\sigma_o} = kg_1$. The value of the scalar k is easily determined from $kg_1 = \dfrac{g_1}{u_1'\sigma_o}$ to be $k^2 = \dfrac{1}{g_1'\sigma_o}$ and C_1 takes the very simple form $C_1 = C_o + \dfrac{g_1g_1'}{g_1'\sigma_o}$. Applying the rank-one inversion formula to C_1 yields

$$D_1 = C_1^{-1} = C_o^{-1} - \frac{C_o^{-1}\frac{g_1 g_1'}{g_1' c_o} C_o^{-1}}{1 + \frac{g_1' C_o^{-1} g_1}{g_1' c_o}}, \text{ or } D_1 = D_o - \frac{D g_1 g_1' D_o}{g_1' c_o + g_1' D_o g_1}.$$

But $g_1' D_o g_1 = g_1' D_o (c_o + y_o)$

$= g_1' D_o y_o - g_1' c_o,$

so $D_1 = D_o - \frac{D_o g_1 g_1' D_o}{g_1' D_o y_o}.$

Just as $C_1 \sigma_o = y_o$, $D_1 y_o = \sigma_o$ for $D_1 y_o = D_o y_o - \frac{D_o g_1 g_1' D_o y_o}{g_1' D_o y_o}$

$= D_o(g_1 - g_o) - D_o g_1 = -D_o g_o = \sigma_o$

or simply by applying $D_1 = C_1^{-1}$ to $C_1 \sigma_o = y_o$.

Since D_1 is the current best estimate of G^{-1}, it is reasonable to continue the process by computing $\sigma_1 = -D_1 g_1$, $x_2 = x_1 + \sigma_1$, $g_2 = \text{grad } f(x_2)$ and $y_1 = g_2 - g_1$ and considering an improved estimate C_2 to G obtained from C_1, g_2 and y_1 in precisely the same manner. More generally suppose x_n, D_n, g_n and C_n have been obtained. Set

$$\sigma_n = -D_n g_n,$$

$$x_{n+1} = x_n + \sigma_n,$$

$$g_{n+1} = \text{grad } f(x_{n+1}),$$

and $$y_n = g_{n+1} - g_n = G\sigma_n.$$

Define $D_{n+1} = D_n - \frac{D_n g_{n+1} g_{n+1}' D_n}{g_{n+1}' D_n y_n}$

and $C_{n+1} = C_n + \frac{g_{n+1} g_{n+1}'}{g_{n+1}' \sigma_n}.$

Clearly $C_{n+1}\sigma_n = C_n \sigma_n + g_{n+1} = -g_n + g_{n+1} = y_n = G\sigma_n$ and

$D_{n+1} y_n = D_n y_n - D_n g_{n+1} = D_n g_{n+1} - D_n g_n - D_n g_{n+1} = \sigma_n = G^{-1} y_n.$

An important property of the algorithm is if $D_n y = \sigma = G^{-1}y$ for an arbitrary y then $D_{n+1}y = D_n y$. This property is demonstrated by showing $g'_{n+1}D_n y = 0$ as follows. Since $g_{n+1} - g_n = G(x_{n+1} - x_n)$,

$$\sigma_n = x_{n+1} - x_n = G^{-1}(g_{n+1} - g_n) = -D_n g_n,$$

or $-D_n g_n - G^{-1}(g_{n+1} - g_n) = 0.$

Then $D_n g_{n+1} = D_n g_n - G^{-1}(g_{n+1} - g_n) + D_n g_{n+1}$

$$= (D_n - G^{-1})(g_{n+1} - g_n),$$

and $g'_{n+1}D_n y = (g_{n+1} - g_n)'(I_n - G^{-1})y = 0.$

That is to say, if D_n agrees with G^{-1} for some vector y, then so does D_{n+1}.

Clearly D_{n+1} cannot be computed if the term $g'_{n+1}D_n y_n$ vanishes. If this occurs as above because D_n agrees with G^{-1} on y_n, then

$D_n y_n = G^{-1}y_n = G^{-1}(g_{n+1} - g_n)$

$= G^{-1}(a + Gx_{n+1} - a - Gx_n)$

$= x_{n+1} - x_n = \sigma_n = -D_n g_n$. But also, $D_n y_n = D_n g_{n+1} - D_n g_n$, therefore $D_n g_{n+1} = 0$ and if D_n is non-singular $g_{n+1} = 0$ which in turn implies x_{n+1} is the desired solution.

By Lemma I, D_{n+1} exists so long as the denominator $(1 + v'A^{-1}u$ in the Lemma) does not vanish. In the case of the D'_ks, D_{n+1} exists so long as $g'_{n+1}\sigma_n + g'_{n+1}D_n g_{n+1} = g'_{n+1}D_n y_n \neq 0$. Furthermore, C_{n+1} can be obtained from D_{n+1} by applying the same inversion technique, hence the denominator in the expression $C_{n+1} = C_n + \dfrac{g_{n+1}g'_{n+1}}{g'_{n+1}\sigma_n}$ must not vanish.

There are then two requirements for the existence of $D_{n+1} = C_{n+1}^{-1}$ and $C_{n+1} = D_{n+1}^{-1}$, namely: $g'_{n+1}D_n y_n \neq 0$ and $g'_{n+1}\sigma_n \neq 0$.

In the following it is assumed, for the moment, that these requirements are satisfied.

The algorithm may be generalized to one for minimizing an arbitrary function f defined on E^N as follows:

Initially: choose an arbitrary x_o, set $D_o = I$

and compute $g_o = \text{grad } f(x_o)$.

Iteratively: Set $\sigma_n = -D_n g_n$ and $x_{n+1} = x_n + \sigma_n$.

Compute $g_{n+1} = \text{grad } f(x_{n+1})$,

set $y_n = g_{n+1} - g_n$ and

$$\text{compute } D_{n+1} = D_n - \frac{D_n g_{n+1} g'_{n+1} D_n}{g'_{n+1} D_n y_n} .$$

As an algorithm for minimizing an arbitrary function, precautions must be taken to avoid instances where the denominator will vanish. In addition, for an arbitrary function, D_n may not be positive definite and hence $\sigma_n = -D_n g_n$ may not be a descent direction. Therefore, additional precautions must be incorporated into the algorithm to assure applicability to arbitrary functions at the expense of increased complexity.

The recursive relation for D_{n+1} may be rewritten as follows:

$$D_{n+1} = D_n - \left(\frac{g'_{n+1} D_n g_{n+1}}{g'_{n+1} D_n y_n}\right) \frac{D_n g_{n+1} g'_{n+1} D_n}{g'_{n+1} D_n g_{n+1}}$$

or $$D_{n+1} = D_n + (\lambda_n - 1) \frac{D_n g_{n+1} g'_{n+1} D_n}{g'_{n+1} D_n g_{n+1}}$$

where $\lambda_n = 1 - \dfrac{g'_{n+1} D_n g_{n+1}}{g'_{n+1} D_n y_n}$.

This form for D_{n+1} is essentially the recursive relation of Davidon's Variance Algorithm except that λ_n is chosen to assure both D_{n+1} and D_{n+1}^{-1} (i.e., C_{n+1}) remain positive definite at each iteration, where the

above relation for λ_n is used whenever possible. Specifically, λ_n is chosen such that $\alpha x'D_n x \leq x'D_{n+1}x \leq \beta x'D_n x$ for all x, where $0<\alpha<1<\beta$.

As Davidon's Variance Algorithm is presented in Reference 3, D_{n+1} (or V^* in Davidon's notation) is constructed from a test value x^* for x_{n+1}. If $f(x^*) \geq f(x_n)$, then x_n is taken for x_{n+1}, that is the estimate x^* is discarded. As pointed out by Davidon in a footnote (Reference 3, p. 406) the algorithm can become trapped in a loop. Indeed, although the poor estimate x^* is not used, the gradient of f at x^* is used to modify the estimate, D_n, of G^{-1}. Since for most problems f(x) can be computed much more rapidly than g(x) = grad f(x), several test computations of f(x), without the corresponding gradient, can be made without undue increase in computation time. The following modification to Davidon's Variance algorithm provides for a search for an improved estimate for x at each iteration, if necessary, using several computations of f(x) before computing g(x) and updating the estimate of G^{-1}. Since this search introduces a major change in the algorithms, the form of the general term is re-derived as follows.

Assume x_n, g_n, $D_n = C_n^{-1}$ have been obtained where g_n = grad $f(x_n)$, D_n is positive definite, and $f(x) = f_o + a'x + 1/2\ x'Gx$. Pick an α_n such that $f(x_n - \alpha_n D_n g_n) < f(x_n)$ using $\alpha_n = 1$ whenever possible, otherwise α_n is reduced (for example $\alpha_n = 1/2,\ 1/4, \ldots$) until f is decreased. This is possible since $[f'(x_n), -D_n g_n] = -g'_n D_n g_n < 0$. Set $\sigma_n = -\alpha_n D_n g_n$, $x_{n+1} = x_n + \sigma_n$ and compute g_{n+1} = grad $f(x_n) = g_n + G\sigma_n$ and $y_n = g_{n+1} - g_n = G\sigma_n$. An improved estimate C_{n+1} to G of the form $C_{n+1} = C_n + u_n u_n'$ is sought such that $C_{n+1}\sigma_n = G\sigma_n = y_n = g_{n+1} - g_n$. As before, $C_{n+1}\sigma_n = C_n\sigma_n + u_n u_n'\sigma_n = g_{n+1} - g_n$ from which $u_n = \dfrac{g_{n+1} - g_n - C_n\sigma_n}{u_n'\sigma_n}$,

or $$u_n = \frac{g_{n+1} - (1 - \alpha_n)g_n}{u_n'\sigma_n}$$

since $C_n\sigma_n = -\alpha_n C_n(D_n g_n) = -\alpha_n g_n$.

For clarity, set $v_n = g_{n+1} - (1 - \alpha_n)g_n$ and observe u_n has the form $u_n = \frac{v_n}{k}$. Then $\sigma_n = \frac{v_n}{k} = \frac{v_n}{u_n'\sigma_n} = \frac{v_n}{v_n'\sigma_n/k}$ and $u_n'\sigma_n = \frac{v_n'\sigma_n}{k} = \frac{v_n'\sigma_n}{v_n'\sigma_n/k} = k$, from which $k^2 = v_n'\sigma_n$.

Now C_{n+1} may be written $C_{n+1} = C_n + \frac{v_n v_n'}{v_n'\sigma_n}$ and an application of the rank-one inversion formula yields $D_{n+1} = C_{n+1}^{-1} = D_n - \frac{D_n v_n v_n' D_n}{v_n'\sigma_n + v_n' D_n v_n}$.

Since $v_n' D_n v_n = v_n' D_n [g_{n+1} - g_n + \alpha_n g_n]$

$= v_n' D_n y_n + \alpha_n v_n' D_n g_n$

$= v_n' D_n y_n - v_n'\sigma_n$,

D_{n+1} may be simplified to $D_{n+1} = D_n - \frac{D_n v_n v_n' D_n}{v_n' D_n y_n}$.

This form of the algorithm can be shown to have the same properties as Davidon's form. For example if D_n agrees with G^{-1} on y, i.e., $D_n y = G^{-1} y = \sigma$, then so does D_{n+1}. This is demonstrated, as before, by showing $v_n' D_n y = 0$ as follows:

$$D_n v_n = D_n(g_{n+1} - g_n + \alpha_n g_n) = D_n(g_{n+1} - g_n) - \sigma_n$$

but $\sigma_n = x_{n+1} - x_n = G^{-1}(g_{n+1} - g_n)$, so $D_n v_n = (D_n - G^{-1})(g_{n+1} - g_n)$, and $v_n' D_n y = (g_{n+1} - g_n)'(D_n - G^{-1})y = 0$.

This form of the algorithm also suffers the difficulty of D_{n+1} not necessarily being positive definite at each iteration for an arbitrary function. This difficulty can be avoided by applying Davidon's method of assuring boundedness. D_{n+1} may be written as

$$D_{n+1} = D_n - \left(\frac{g_{n+1}' D_n g_{n+1}}{v_n' D_n y_n}\right) \frac{D_n v_n v_n' D_n}{g_{n+1}' D_n g_{n+1}}$$

$$\text{or } D_{n+1} = D_n + (\lambda_n - 1)\frac{D_n v_n v_n' D_n}{g_{n+1}' D_n g_{n+1}}$$

$$\text{where} \quad \lambda_n = 1 - \frac{g_{n+1}' D_n g_{n+1}}{v_n' D_n y_n} .$$

Following Davidon, λ_n is chosen by the above relation whenever possible, otherwise such that $\alpha \leq \lambda_n \leq \beta$ where $0<\alpha<1<\beta<\infty$ in order to assure

$$\alpha x' D_{n+1} x \leq x' D_n x \leq \beta x' D_n x$$

for any x.

3. COMPLETE RANK-ONE ALGORITHM FOR FUNCTION MINIMIZATION

The complete algorithm proceeds as follows:

(1) Initially set $D_o = I$ and pick α, β such that $0<\alpha<1<\beta$.

Choose an arbitrary x_o and compute $f(x_o)$ and $g(x_o) = \text{grad } f(x_o)$.

(2) Compute $f(x_n - \alpha_n D_n g_n)$ for $\alpha_n = 1, 1/2, 1/4, \ldots$ until

$f(x_n - \alpha_n D_n g_n) < f(x_n)$

(3) Set $\sigma_n = -\alpha_n D_n g_n$ and $x_{n+1} = x_n + \sigma_n$

Compute $g_{n+1} = g(x_{n+1}) = \text{grad } f(x_{n+1})$

and set $y_n = g_{n+1} - g_n$,

$$v_n = y_n + \alpha_n g_n = g_{n+1} - (1 - \alpha_n) g_n$$

and $$\gamma_n = \frac{g_{n+1}' D_n g_{n+1}}{v_n' D_n y_n}$$

(4) If $\gamma_n > 1 - \alpha$ set $\lambda_n = \alpha$,

if $\gamma_n < 1 - \beta$ set $\lambda_n = \beta$,

otherwise set $\lambda_n = 1 - \gamma_n$.

(5) Set $D_{n+1} = D_n + (\lambda_n - 1) \frac{D_n v_n v_n' D_n}{g_{n+1}' D_n g_{n+1}}$

(6) Repeat steps (2) through (5) until selected error tolerances(s) is (are) satisfied. One or more of the following error tests may be used:

$$||g_{n+1}|| < \varepsilon_1,\ g_{n+1}' D_n g_{n+1} < \varepsilon_2,$$

$$f(x_n) - f(x_{n+1}) < \varepsilon_3, \text{ and } ||\sigma_n|| < \varepsilon_4.$$

It is interesting to note that any algorithm which generates recursive estimates to the inverse of the Hessian as do the Fletcher-Powell algorithm, Davidon's Variance algorithm, or the modified Davidon algorithm presented above can be expected to exhibit numerical difficulties whenever the Hessian or its inverse is singular at the minimizing point. The following examples illustrate simple problems with this property.

Consider the problem of minimizing $f(x,y) = x^2 + y^4$. The first and second derivatives are $\nabla f(x,y) = \begin{pmatrix} 2x \\ 4y^3 \end{pmatrix}$ and

$$\nabla^2 f(x,y) = \begin{bmatrix} 2 & 0 \\ 0 & 12y^2 \end{bmatrix}.$$

Clearly the minimum is at $(x,y) = (0,0)$ where $\nabla f(x,y) = (0,0)'$ but $\nabla^2 f(0,0) = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$ is singular. Also,

$$\nabla^2 f(x,y)^{-1} = \begin{bmatrix} 1/2 & 0 \\ 0 & \frac{1}{12y^2} \end{bmatrix} \xrightarrow[y \to 0]{} \begin{bmatrix} 1/2 & 0 \\ 0 & \infty \end{bmatrix}.$$

Any algorithm which attempts to estimate $\nabla^2 f(x,y)^{-1}$ can well be expected to have terms which tend to become unbounded as the solution is approached.

For an example of the inverse condition, consider minimizing $f(x,y) = x^2 + y^{4/3}$. Then, formally, $\nabla f(x,y) = \begin{pmatrix} 2x \\ 4/3y^{1/3} \end{pmatrix}$ and

$$\nabla^2 f(x,y) = \begin{bmatrix} 2 & 0 \\ 0 & \frac{4}{9} y^{-2/3} \end{bmatrix}.$$

Clearly the minimum is at $(x,y) = (0,0)$ and in this case

$$\nabla^2 f(x,y)^{-1} = \begin{bmatrix} 1/2 & 0 \\ 0 & \frac{4}{9} y^{2/3} \end{bmatrix} \xrightarrow[y \to 0]{} \begin{bmatrix} 1/2 & 0 \\ 0 & 0 \end{bmatrix}.$$

In this case, any algorithm which attempts to generate $\nabla^2 f(0,0)^{-1}$ can be expected to become singular.

In the usual formulation of the Fletcher-Powell algorithm the matrix H_r is tested at each iteration and if any elements become too large or too small the algorithm is re-started. Thus, in the worse case, the Fletcher-Powell algorithm would de-generate into the usual gradient algorithm. Davidon's algorithm, both the original and as modified here, provide assurance (through α,β) that the computations can proceed by making conservative estimates to the inverse of the Hessian.

The relative speed of Davidon's algorithm compared to conjugate gradient and Fletcher-Powell algorithms was reported on in Reference 11 . Those results indicate the Davidon method is the superior algorithm for the solution of most of the test problems considered. In the few cases where another algorithm was found superior the differences were either marginal or, as in a few cases, the Fletcher-Powell algorithm arrived at an exact solution.

SECTION IV

DIRECT APPLICATION OF RANK-ONE

Although rank-one methods have been used to construct algorithms for minimizing arbitrary functions, the direct application of rank-one to minimizing special classes of functions has been overlooked. Consider, for example, minimizing the function

$$f(x) = 1/2||x||^2 + 1/2k(a - m'x)^2$$

where $x, m \in R^n$ and k, a are scalars. The gradient of f at x is

$$\nabla f(x) = x + k(a - m'x)(-m).$$

The necessary condition for f to be minimum at $x = x^*$ is that the gradient ($\nabla f(x^*)$) be zero. Furthermore, since m'x is a scalar, (m'x)m can be rewritten as m(m'x) = (mm')x, where mm' is the outer or tensor product, i.e., an nxn rank-one matrix. Use this fact and set the gradient to zero to obtain

$$x^* - kam + kmm'x^* = 0$$

or

$$(I + kmm')x^* = kam.$$

The rank-one matrix inversion technique (Section III) is applied to obtain directly

$$x^* = (I + kmm')^{-1}kam$$

$$= (I - \frac{kmm'}{1 + km'm})kam$$

$$= \frac{ka}{1 + km'm} m.$$

More complex forms can be handled with little difficulty. Let M be a m x n matrix, a an m-vector and consider minimizing

$$f(x) = 1/2||x||^2_{R^n} + 1/2k||a - Mx||^2_{R^m}$$

$$= 1/2x'x + 1/2k(a - Mx)'(a - Mx).$$

The gradient of this function is given by

$$\nabla f(x) = x + k(-M')(a - Mx) = x - kM'a + kM'Mx.$$

Again, for a minimum of f at x^*

$$\nabla f(x^*) = (I + kM'M)x^* - kM'a = 0$$

or

$$x^* = (I + kM'M)^{-1}kM'a$$

To clarify the structure of M'M, let m_i be the $i^{\underline{th}}$ row of M represented as a column vector, m_{ij} the $j^{\underline{th}}$ element of m_i as well as the $i,j^{\underline{th}}$ element of M.

Since $\{M'M\}_{ij} = \sum_{k=1}^{m} m_{ki}m_{kj}$,

and

$$m_k m_k' = \begin{bmatrix} m_{k1} \\ m_{k2} \\ \cdots \\ m_{kn} \end{bmatrix} \begin{bmatrix} m_{k1}, m_{k2}, \cdots, m_{kn} \end{bmatrix}$$

$$= \begin{bmatrix} m_{k1}m_{k1} & m_{k1}m_{k2} & \cdots & m_{k1}m_{kn} \\ m_{k2}m_{k1} & m_{k2}m_{k1} & \cdots & m_{k2}m_{kn} \\ \cdots & & & \\ m_{kn}m_{k1} & m_{kn}m_{k2} & \cdots & m_{kn}m_{kn} \end{bmatrix}$$

it follows that

$$M'M = \sum_{k=1}^{m} m_k m_k' .$$

Now $I + kM'M$ may be represented as $I + \sum_{i=1}^{m} km_i m_i'$ and its inverse may be computed recursively by repeated application of the rank-one inversion

method. The question of the existence of $(I + kM'M)^{-1}$ is answered in Theorem II which follows shortly. Formally, the method of computing $(I + kM'M)^{-1}$ is derived recursively as follows:

Set $C_o = I$ $\qquad D_o = C_o^{-1} = I$

$$C_1 = I + km_1m_1' \qquad D_1 = C_1^{-1} = D_o - \frac{kD_om_1m_1'D_o}{1 + km_1'D_om_1}$$

. . .

$$C_i = C_{i-1} + km_im_i' \qquad D_i = C_i^{-1} = D_{i-1} - \frac{kD_{i-1}m_im_i'D_{i-1}}{1 + km_i'D_{i-1}m_i}$$

. . .

$$C_m = I + kM'M \qquad D_m = C_m^{-1} = (I + kM'M)^{-1}$$

Note D_m is computed in precisely m steps (m = row order of M) and the C_i's need not be computed since D_i can be computed from D_{i-1}, m_i and k. For this problem, $x^* = D_m(kM'a)$ is the required solution satisfying $\nabla f(x^*) = 0$.

1. APPLICATION TO AN AIRCRAFT WING-ROOT BENDING PROBLEM

The above rank-one method was used to determine the optimum aircraft wing control surface deflections required to minimize a specified penalty function (References 8,9). The objective of the problem was to reduce wing-root bending moments through active control of trailing edge control surfaces on the wing. In a physical application such reduced bending moment loads could lead to reduced structural requirements, thus reducing the aircraft weight and improving aircraft performance, or alternatively lead to an expansion of the aircraft operational envelope. An immediate consequence of the control surface deflections is to change other important aircraft wing characteristics, principally lift, pitching moment

and drag. These also affect aircraft performance.

For most applications, increased lift at a given angle-of-attack is desirable because an increase in vertical acceleration can be obtained with less increase in wing angle-of-attack. This leads to less drag and hence more efficient operation.

Any change in wing-generated pitching moment would require changes in aircraft trim to generate balancing moments. Although trim changes can be accomplished automatically, this adds system complexity and interface problems. The desire to avoid complexity is motivated by the usually valid supposition that the more complex system is less reliable.

From drag considerations and because linear aerodynamic theory was used, it was desired to restrict the magnitude of control surface deflections. The use of linear aerodynamic theory, in addition to being conceptually and computationally simpler, was necessary to provide the linear system description.

The mathematical model of this problem was formulated as follows: Changes in wing lift, pitching moment and root bending can be represented, assuming linear aerodynamics, as a linear function of the control surface deflections; therefore let $\delta = (\delta_1, \delta_2, \ldots, \delta_n)$ represent the deflections of the wing control surfaces, $\Delta C_L = m_1'\delta$ the change in wing lift, $\Delta C_M = m_2'\delta$ be the change in wing pitching moment, $\Delta C_{RB} = m_3'\delta$ the change in wing-root bending moment. The following cost function was formulated which when minimized would tend to minimize wing-root bending moment while holding changes in pitching moment small, maximize the change in lift and maintain reasonable control surface deflections:

$$J(\delta) = K_1(\Delta C_{RB}) + K_2(\Delta C_M)^2 - K_3(\Delta C_L) + 1/2\|\delta\|^2$$

where the weighting factors, K_1, K_2, and K_3, are assumed known *a priori*. The solution to this problem is readily obtained by a single application of the rank-one method.

Although the solutions are valid the results are difficult to interpret due to the fact that each solution represents a different wing-lift condition. The problem was reformulated at a constant lift condition. Specifically, an increment in angle-of-attack was incorporated to allow the wing to generate the same lift with or without controls deflected. The more complete representations for the changes in wing characteristics are:

$$\Delta C_L = m_1'\delta + C_{L_\alpha}\alpha,$$

$$\Delta C_M = m_2'\delta + C_{M_\alpha}\alpha,$$

$$\Delta C_{RB} = m_3'\delta + C_{RB_\alpha}\alpha.$$

Thus for constant lift $\Delta C_L = 0$ and $\alpha = -\frac{1}{C_{L_\alpha}} m_1'\delta$.

Then the changes in pitching moment and wing-root bending at constant lift are:

$$\Delta C_M = m_2'\delta - \frac{C_{M_\alpha}}{C_{L_\alpha}} m_1'\delta = \bar{m}_2'\delta$$

$$\Delta C_{RB} = m_3'\delta - \frac{C_{RB_\alpha}}{C_{L_\alpha}} m_1'\delta = \bar{m}_3'\delta .$$

Finally, from linear aerodynamic theory, the induced drag of a wing is minimum when the wing pressure distribution is elliptical. Therefore, an additional term was included to represent the deviation of

the predicted pressure distribution from elliptical. Specifically, the spanwise distribution as a function of the normalized semispan η was represented as $c_{\ell}(\eta) = c_{\ell_c}(\eta) + c_{\ell_\alpha}(\eta)\alpha + c_{\ell_\delta}(\eta)\delta$ and the elliptical distribution as $c_{\ell_{ellip}}(\eta) = \frac{2C_L}{\pi}\sqrt{1-\eta^2}$. Following the above procedure for correcting the angle-of-attack change,

$$c_{\ell_{ellip}}(\eta) - C_{\ell}(\eta) = \left[\frac{2C_L}{\pi}\sqrt{1-\eta^2} - c_{\ell_o}(\eta)\right] - \left[c_{\ell_\delta}(\delta) - \frac{c_{\ell_\alpha}(\eta)}{C_{L_\alpha}} m_1'\right]\delta$$

This is discretized for k spanwise locations (η_i, i=1,. . .,k) then a measure of the spanwise distribution deviation from elliptical is given by $||\lambda_o - \Lambda\delta||^2$, where

$$\{\lambda_o\}_i = \frac{2C_L}{\pi}\sqrt{1-\eta_i^2} - c_{\ell_o}(\eta_i)$$

$$\{A\}_{ij} = c_{\ell_{\delta_j}}(\eta_i) - \frac{c_{\ell_\alpha}(\eta_i)}{C_{L_\alpha}} m_{1j}$$

and the norm is in R^k.

Finally, the following cost function was formulated to minimize wing-root bending, change in pitching moment, control surface deflections and spanwise distribution error:

$$J(\delta) = K_1(\Delta C_{RB}) + K_2(\Delta C_M)^2 + 1/2||\delta||^2_{R^n} + K_5||\lambda_o - \Lambda\delta||^2_{R^k}$$

$$= K_1\bar{m}_2'\delta + K_3(\bar{m}_3'\delta)^2 + 1/2\delta'\delta + K_5(\lambda_o - \Lambda\delta)'(\lambda_o - \Lambda\delta).$$

The minimum of this function is obtained by computing the gradient and equating the gradient to zero. The gradient is given by

$$\nabla J(\delta) = K_1\bar{m}_2 + K_3\bar{m}_3\bar{m}_3'\delta + \delta + K_5(-A')(\lambda_o - A\delta)$$

$$= (I + K_3\bar{m}_3\bar{m}_3' + K_5A'A)\delta + K_1\bar{m}_2 - K_5A'\lambda_o$$

and the value of δ for which $\nabla J(\delta) = 0$ is given by

$$\delta = (I + K_3\bar{m}_3\bar{m}_3' + K_5A'A)^{-1}(K_5A'\lambda_o - K_1\bar{m}_2).$$

The minimizing δ can then be obtained by k+1 iterations of the rank-one method.

In both of the above formulations of the problem the weighting factors, K_i's in the cost function are presumed known. As is often the case in this type problem, this is generally not true. However, the solution to the problem for a given set of K_i's can be computed extremely rapidly on a digital computer using the rank-one method, making a systematic search over a wide variation of the weighting factors a practical approach to obtaining a realistic solution. For example, a program written for the CDC 6600 computer to solve the above problem computes the basic matrices and vectors required (A,m_i's, etc.), increments the K_i's in a systematic fashion and computes over 3,000 optimal δ's in approximately 10.0 seconds central processor, CP, time. With such a volume of data, care had to be taken in the search pattern and output format to avoid an overwhelming output volume.

2. DERIVATION OF THE DIRECT RANK-ONE METHOD

In the preceding development only the solution of the necessary condition for an extremum, $\nabla f(x^*) = 0$, was considered. For the simplest form, a linear combination of quadratic terms with positive coefficients, if an extremum exists it must be a minimum. The addition of linear terms

causes no particular difficulty since the quadratic terms can be expected to eventually dominate. For the more general problem where the weighting factors are arbitrary, the rank-one method also provides some information on the nature of the solution.

For any of the above examples the gradient is represented by the general form $\nabla f(x) = Cx + a$, where C is a matrix and a a vector. The second derivative is $\nabla^2 f(x) = C$. The following Lemma provides information on the positive definiteness of C.

Lemma II

If C_n and $D_n = C_n^{-1}$ are positive definite,

$$D_{n+1} = C_{n+1}^{-1} = (C_n + km_n m_n')^{-1} = D_n - \frac{kD_n m_n m_n' D_n}{1 + km_n' D_n m_n}$$

and $1 + km_n' D_n m_n > 0$, then C_{n+1} and D_{n+1} are positive definite.

Proof

For an arbitrary $u, u'D_n u > 0$ therefore the ratio

$$\frac{u'D_{n+1}u}{u'D_n u} = 1 - \frac{k}{u'D_n u} \frac{u'D_n m_n m_n' D_n u}{1 + km_n' D_n m_n}$$

$$= 1 - \frac{km_n' D_n m_n}{1 + km_n' D_n m_n} \frac{(u'D_n m_n)^2}{(u'D_n u)(m_n' D_n m_n)} .$$

By the Cauchy-Schwarz inequality $(u'D_n m_n)^2 \leq (u'D_n u)(m_n' D_n m_n)$ where equality holds if and only if u is a scalar multiple of m_n. Clearly the ratio $\frac{u'D_{n+1}u}{u'D_n u} = 1$ if $u'D_n m_n = 0$ and this ratio is furthest from one when u is a multiple of m_n, in which case

$$\frac{u'D_{n+1}u}{u'D_nu} = 1 - \frac{km_n'D_nm}{1 + km_n'D_nm_n}$$

$$= \frac{1}{1 + km_n'D_nm_n} .$$

Let $\alpha = \text{Min} \left\{ 1, (1 + km_n'D_nm_n)^{-1} \right\}$

and $\beta = \text{Max} \left\{ 1, (1 + km_n'D_nm_n)^{-1} \right\}$

Then $\alpha u'D_nu \leq u'D_{n+1}u \leq \beta u'D_nu$

Also, for $C_{n+1} = C_n + km_nm_n'$,

$$\frac{v'C_{n+1}v}{v'C_nv} = 1 + \frac{k(v'm_n)^2}{v'C_nv}$$

$$= 1 + km_n'D_nm_n \frac{(v'm_n)^2}{(v'C_nv)(m_n'D_nm_n)} .$$

Since $C_n = D_n^{-1}$ and C_n and D_n are positive definite

$$(v'm_n)^2 \leq (v'C_nv)(m_n'D_nm_n),$$

where equality holds if and only if v is a scalar multiple of m_n. Thus the ratio $\frac{v'C_{n+1}v}{v'C_nv}$ is one if $v'm_n = 0$ and furthest from one when v is a multiple of m_n in which case

$$\frac{v'C_{n+1}v}{v'C_nv} = 1 + km_n'D_nm_n .$$

Let $\gamma = \text{Min} \left\{ 1, 1 + km_n'D_nm_n \right\}$

and $\lambda = \text{Max}\left\{1,\ 1 + km_n' D_n m_n\right\}$

then $\qquad \gamma v' C_n v \le v' C_{n+1} v \le \lambda\, v' C_n v.$

Thus the denominator, $1 + km_n' D_n m_n$, provides a necessary and sufficient condition for the invertibility of C_{n+1}, i.e., the existence of D_{n+1}, by Lemma I and a sufficient condition for C_{n+1} and D_{n+1} to be positive definite by Lemma II.

Before proceeding to a proof of the rank-one method, the following observations are made to simplify the notation. For the problem of minimizing

$$f(x) = ||x||^2_{R^n} + k_1||a - Mx||^2_{R^m} + k_2||\lambda_o - \Lambda x||^2_{R^p},$$

the gradient is given by

$$\begin{aligned} \nabla f(x) &= 2x + 2k_1(-M')(a - Mx) \\ &\quad + 2k_2(-\Lambda')(\lambda_o - \Lambda x) \\ &= 2(I + k_1 M'M + k_2 \Lambda'\Lambda)x - 2k_1 M'a - 2k_2 \Lambda'\lambda_o. \end{aligned}$$

If x^* minimizes f then $\nabla f(x^*) = 0$ and if $(I + k_1 M'M + k_2 \Lambda'\Lambda)^{-1}$ exists, then $x^* = (I + k_1 M'M + k_2 \Lambda'\Lambda)^{-1}(k_1 M'a + k_2 \Lambda'\lambda_o)$. Furthermore, if $(I + k_1 M'M + k_2 \Lambda'\Lambda)$ is positive definite then f has a local minimum at x^*.

As before let m_i, $i = 1,2,\ldots,m$, be the $i^{\underline{th}}$ row of M and λ_i, $i=1,\ldots,p$, be the $i^{\underline{th}}$ row of Λ. Then $(I + k_1 M'M + k_2 \Lambda'\Lambda)$ can be written as

$$I + \sum_{i=1}^{N} \eta_i v_i v_i'$$

where $N = m + p$

$$\eta_i = \begin{cases} k_1 & i = 1,\ldots,m \\ k_2 & i = m+1,\ldots,m+p \end{cases}$$

$$v_i = \begin{cases} m_i & i = 1,\ldots,m \\ \lambda_{i-m} & i = m+1,\ldots,m+p \end{cases}.$$

Define C_k and D_k recursively for $k = 0,1,\ldots,N$ as follows:

$$C_o = I \qquad D_o = C_o^{-1} = I$$

$$\cdots \qquad \cdots$$

$$C_k = C_{k-1} + \eta_k v_k v_k' \qquad d_k = 1 + \eta_k v_k' D_{k-1} v_k$$

$$D_k = D_{k-1} - \frac{\eta_k}{d_k} D_{k-1} v_k v_k' D_{k-1}.$$

Theorem II

With C_k, d_k and D_k as defined above

(a) $C_N = I + \sum_{i=1}^{N} \eta_i v_i v_i'$

(b) so long as $d_i \neq 0$, $i = 1,\ldots,k$

$D_k = C_k^{-1}$ and in particular for $k = N$,

$$D_N = C_N^{-1} = (I + \sum_{i=1}^{N} \eta_i v_i v_i')^{-1}$$

(c) if $d_k > 0$ for all $k = 1,2,\ldots,N$, D_N is positive definite.

Proof

(a) By construction:

$C_o = I$,

$C_1 = I + \eta_1 v_1 v_1'$,

$$C_2 = C_1 + \eta_2 v_2 v_2'$$

$$= I + \eta_1 v_1 v_1' + \eta_2 v_2 v_2'$$

$$= I + \sum_{i=1}^{2} \eta_i v_i v_i',$$

...

$$C_k = I + \sum_{i=1}^{k} \eta_i v_i v_i'$$

...

$$C_N = I + \sum_{i=1}^{N} \eta_i v_i v_i'.$$

(b) By induction:

Initially $C_o = I$ and $D_o = C_o^{-1} = I$. Suppose the assertion holds for k-1, i.e., $D_{k-1} = C_{k-1}^{-1}$ and $d_i \neq 0$, for i=1,...,k-1. Now if $d_k \neq 0$, D_k exists by Lemma I and

$$C_k D_k = (C_{k-1} + \eta_k v_k v_k')(D_{k-1} - \frac{\eta_k}{d_k} D_{k-1} v_k v_k D_{k-1})$$

$$= C_{k-1} D_{k-1} - \frac{\eta_k}{d_k} C_{k-1} D_{k-1} v_k v_k' D_{k-1}$$

$$+ \eta_k v_k v_k' D_{k-1} - \frac{\eta_k^2}{d_k} v_k v_k' D_{k-1} v_k v_k' D_{k-1}.$$

Since $C_{k-1} D_{k-1} = I$ and $v_k' D_{k-1} v_k$ is a scalar, the above may be rewritten as

$$C_k D_k = I - \frac{\eta_k}{d_k} v_k v_k' D_{k-1} + \eta_k v_k v_k' D_{k-1}$$

$$- \frac{\eta_k^2}{d_k} (v_k' D_{k-1} v_k) v_k v_k' D_{k-1}$$

$$= I - \frac{\eta_k}{d_k} v_k v_k' D_{k-1} (1 - d_k + \eta_k v_k' D_{k-1} v_k)$$

$$= I.$$

Where $d_k = 1 + \eta_k v_k' D_{k-1} v_k$ is used. Thus the assertion holds for k. Furthermore, if $d_k \neq 0$ for k=1,...,N then $D_k = C_k^{-1}$ for k=1,...,N and in particular, from (a),

$$D_N = C_N^{-1} = (I + \sum_{i=1}^{N} \eta_i v_i v_i')^{-1}$$

(c) By induction:

Clearly $D_o = C_o^{-1} = I$ is positive definite.

Suppose the assertion holds for k-1, i.e., D_{k-1} is positive definite. If $d_k > 0$ then by Lemma II the ratio $\frac{u'D_k u}{u'D_{k-1}u}$ lies between 1 and $d_k^{-1} > 0$ for arbitrary u. Hence, $u'D_k u$ is positive for all u which im- D_k is positive definite and the assertion holds for k. If $d_k > 0$ for all k = 1,...,N then D_N is positive definite.

It should be noted that the matrices C_k need not be computed since D_k and d_k can be computed from D_{k-1}, v_k and η_k directly. The denominator d_k provides a convenient check: if d_k is zero (or sufficiently small) then C_k is singular and D_k fails to exist (D_k may be numerically untractable); if d_k is positive at every iteration then the resultant extremum is the minimum since D_N is positive definite. However, if d_k is negative for one or more iterations, D_N may still be positive definite since Lemma II provides only a sufficient condition. In fact it may happen that $I + \sum_{i=1}^{N} \eta_i v_i v_i'$ is invertible, and even positive definite, however an intermediate D_k may fail to exist because d_k is zero. In

such a case, a reordering of the steps will correct the difficulty as shown below.

Suppose all steps up to the $n^{\underline{th}}$, $n<N$, have been accomplished and $d_n = 1 - \eta_n v_n' D_{n-1} v_n = 0$. Then D_n fails to exist. The role of $\eta_n v_n v_n'$ and $\eta_{n+1} v_{n+1} v_{n+1}'$ may be interchanged, then $\bar{d}_n = 1 + \eta_{n+1} v_{n+1}' D_{n+1} v_{n+1}$ is computed and presumed non-zero. Then

$$\bar{D}_n = D_{n-1} - \frac{\eta_{n+1}}{\bar{d}_n} D_{n-1} v_{n+1} v_{n+1}' D_{n-1}$$

exists and is the inverse of $\bar{C}_n = C_{n-1} + \eta_{n+1} v_{n+1} v_{n+1}'$.

Now the term $\eta_n v_n v_n'$ is again considered

$$\begin{aligned}
\bar{d}_{n+1} &= 1 + \eta_n v_n' \bar{D}_n v_n \\
&= 1 + \eta_n v_n' (D_{n-1} - \frac{\eta_{n+1}}{d_n} D_{n-1} v_{n+1}' D_n) v_n \\
&= 1 + \eta_n v_n' D_{n-1} v_n - \frac{\eta_n \eta_{n+1}}{d_n} v_n' D_{n-1} v_{n+1} v_{n+1}' D_n v_n \\
&= - \frac{\eta_n \eta_{n+1}}{\bar{d}_n} (v_n' D_{n-1} v_{n+1})^2,
\end{aligned}$$

since the first two terms form $d_n = 0$. In general $d_{n+1} \neq 0$ and hence $D_{n+1} = \bar{D}_n - \frac{\eta_n}{\bar{d}_{n+1}} \bar{D}_n v_n v_n' \bar{D}_n$ exists.

A simple numerical example is presented to illustrate this special case. Consider the problem of inverting $I - \binom{1}{0}(1,0) + \binom{1}{1}(1,1)$. Proceeding as usual, set $D_o = I$ and compute $d_1 = 1 + (-1)(1,0)\binom{1}{0} = 1 - 1 = 0$.

Since D_1 fails to exist, the role of (1,0) and (1,1) are interchanged and $\bar{d}_1$ is computed as $\bar{d}_1 = 1 + (1)(1,1)\binom{1}{1} = 1 + 2 = 3$.

Therefore, $\bar{D}_1 = I - \frac{1}{3}\binom{1}{1}(1,1) = \begin{bmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{bmatrix}$.

Now, picking up the first term, $\bar{d}_2 = 1 + (-1)(1,0)\bar{D}_1\binom{1}{0} = 1 - \frac{2}{3} = \frac{1}{3}$

and $D_2 = \bar{D}_1 - \frac{(-1)}{1/3}\bar{D}_1\binom{1}{0}(1,0)\bar{D}_1$

$$= \begin{bmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{bmatrix} + 3\binom{2/3}{-1/3}(2/3,\ -1/3)$$

$$= \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}.$$

The correctness of the solution is readily verified as the inverse of

$$I - \binom{1}{0}(1,0) + \binom{1}{1}(1,1) = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}.$$

SECTION V

CONCLUSIONS

The main topic of this dissertation, computational methods for the solution of unconstrained minimization problems, is covered in three parts; generalized descent algorithms, a rank-one minimization technique for the general problem, and a direct rank-one method for a special class of problems. Although each topic relates to unconstrained minimization, the scope is successively decreased as the specialization is increased.

The theorem on generalized descent algorithms, Theorem I, demonstrates the essential properties of a descent algorithm. Here a descent algorithm is defined to be a computational method in which, at each iteration, a descent direction is generated and a single-dimensional search is conducted for a minimum. Since the setting is highly abstract, additional specialization is included and the theorem is applied to the three descent algorithms in common use: the gradient method, conjugate gradients and the Fletcher-Powell method. The essential property of descent algorithms, choosing a descent direction and the search for a minimum at each iteration, is sufficient to cause the sequence of function values to monotonically decrease. An additional property is required to assure the derivative of t e function goes to zero.

The descent direction generated by the algorithm must be strict in the sense that convergence of the inner product $[f'(x),\phi(x)]$ to zero must imply convergence of the derivative to zero. This condition is made rigorous in condition (ii) required of the function ϕ. This condition and the uniform continuity of the derivative f' are used to demonstrate the derivative must converge to zero.

Although Theorem I demonstrates the common properties of descent algorithms, it fails to provide any information on one very important property of all such algorithms, convergence rate. Convergence rate involves not only the amount of improvement at each iteration but also, for very practical reasons, the computational time required to accomplish the iterations.

One of the common properties of descent algorithms, the single-dimensional search can also be a drawback in that this is generally the most time consuming step of each iteration. Therefore an algorithm which minimizes a function without requiring repeated single-dimensional searches for a minimum might be superior to any descent method. One such method, Davidon's Variance Algorithm (Reference 13), has been shown to be, in many cases, superior to the three common descent algorithms (Reference 11). Davidon's method is discussed in Section III.

A new derivation of Davidon's method is presented. This derivation provides a clearer insight into the structure of the algorithm by considering the algorithm as repeated applications of the rank-one matrix inversion technique. Although the rank-one method would be exact when applied to a quadratic form and would require no single-dimensional search, certain added computational precautions must be included if the method is to be used to minimize an arbitrary function. One of these is provided by Davidon and assures the matrix approximation to the inverse of the Hessian is well behaved. Another precaution provided by this author assures only "good" estimates of the solution are used to update the approximation of the inverse of the Hessian. Although a linear search is involved, only the function value is computed and the function need only be decreased not minimized. As a result, for each iteration

both the number of test points is reduced and the complexity of computations at each test point is reduced.

An extension of the Davidon algorithm to the problem of minimizing an arbitrary function defined on an infinite dimensioned space such as a functior space is highly interesting. In the finite dimensional case the outer product of vectors, uv', can be represented as a matrix. If u and v belong to a function space, the representation of uv' is not entirely obvious. However, since most problems require digital computation, and the representation of a function is necessarily discretized, the function space may be considered as R^N with N very large.

The third area considered in this dissertation applies the rank-one method to a special class of problems. Although the class of problems to which the method applies is specialized, it is not uncommon. In fact many preliminary engineering problems are in this class where the cost function is a combination of linear and quadratic terms each weighted by a penalty factor which is constant but unknown. The rank-one method provides a rapid solution to the problem. The method also provides necessary and sufficient tests for the existence of an extremum and a sufficient test for the solution to be a minimum.

Extension of the direct rank-one method to problems in a function space is of continuing interest to the author. As noted for the extension of Davidon's method, care must be observed in the interpretation of the outer product. Since the number of iterations of rank-one required to obtain the solution is determined by the structure of the cost function to be minimized, not the dimension of the underlying vector space, application of the method in a function space has possibilities.

To date the class of problems in a function space which may be solved by the rank-one method is very restricted and the set of known applications is empty. Finally, the problem of representing functions on a digital computer still exists.

To summarize, the following extensions appear to be promising areas for further investigation. Application of Theorem I to other descent algorithms. Application of Davidon's rank-one method to a large sample of test problems to determine in a practical application its computational speed. Application of Davidon's method in a function space might be fruitful if first the direct rank-one method can be successfully applied.

REFERENCES

1. F. S. Beckman, "The Solution of Linear Equations by the Conjugate Gradient Method", Mathematical Methods for Digital Computers, A. Ralston and H. S. Wilf (Eds.), John Wiley & Sons, Inc., New York, N. Y., 1960.

2. W. C. Davidon, Variable Metric Method for Minimization, Argonne National Lab., Report 5990 (Rev.), 1959.

3. W. C. Davidon, "Variance Algorithm for Minimization", Computer J., 10 (1968), pp. 406-410.

4. R. Fletcher and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization", Computer J., 6 (1963), pp. 163-168.

5. R. Fletcher and C. M. Reeves, "Function Minimization by Conjugate Gradients", Computer J., 7 (1964), pp. 149-154.

6. A. A. Goldstein, "On Steepest Descent", J. SIAM Control, Ser. A, Vol. 3, No. 1, 1965, pp. 147-151.

7. A. A. Goldstein, "Minimizing Functionals on Normed Linear Spaces", J. SIAM Control, Vol. 4, No. 1, 1966, pp. 81-89.

8. J. E. Jenkins, D. C. Eckholdt, and B. T. Kujawski, "An Assessment of the Interfacing Problems with CCV Design Concepts", AIAA 2nd Aircraft Design and Operations Meeting, July 1970, AIAA Paper No. 70-926.

9. B. T. Kujawski, J. E. Jenkins, and D. C. Eckholdt, "Longitudinal Analysis of Two CCV Design Concepts", AIAA 3rd Aircraft Design and Operations Meeting, July 1971, AIAA Paper No. 71-786.

10. L. S. Lasdon, S. K. Mitter, and A. D. Warren, "The Conjugate Gradient Method for Optimal Control Problems", IEEE Trans, on Automatic Control, Vol. AC-12, Nr. 2, April 1967, pp. 132-138.

11. Craig E. Miller, A Computational Comparison of Gradient Minimization Algorithms, Masters Thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, March 1971.

12. M. J. D. Powell, "A Survey of Numerical Methods for Unconstrained Optimization", SIAM Review, Vol. 12, Nr. 1, (1970) pp. 79-97.

13. A. E. Taylor, Advanced Calculus, Ginn and Company, Boston, Mass., 1955.

14. M. M. Vainberg, Variational Methods for the Study of Nonlinear Operators, Holden-Day, Inc., 1964.

15. H. S. Wilf, "Matrix Inversion by the Annihilation of Rank", J. SIAM, Vol. 7, Nr. 2, June 1959, pp. 149-151.